

Pemrograman *Prototype* Penstabil Tegangan Untuk Mengatasi Gangguan *Over-Under Voltage* Menggunakan Arduino

Ramadhan Firdiyanto Firdaus

D3 Teknik Listrik, Fakultas Teknik, Universitas Negeri Surabaya
ramadhan.17050413001@mhs.unesa.ac.id

Subuh Isnur Haryudo, Achmad Imam Agung, Aditya Chandra Hermawan

D3 Teknik Listrik, Fakultas Teknik, Universitas Negeri Surabaya
subuhisnur@unesa.ac.id, achmadimam@unesa.ac.id, adityahermawan@unesa.ac.id

Abstrak

Gangguan kelistrikan pada konsumen mengakibatkan terjadinya penghambatan pada kegiatan manusia sehari-hari. Gangguan yang terjadi ialah dari berbagai aspek. Salah satu gangguan yang terjadi adalah gangguan *Over-Under Voltage* pada instalasi daya. Gangguan ini jika terhubung pada peralatan listrik akan mempengaruhi kinerja peralatan listrik tersebut dan mengurangi umur peralatan tersebut. Untuk itu maka dibuatlah *Prototype* Penstabil Tegangan Untuk Mengatasi Gangguan *Over-Under Voltage* Pada Instalasi Daya Menggunakan Arduino. Alat ini menunjukkan 3 kondisi yaitu *under voltage* dimana tegangan dibawah 220V, *normal voltage* tegangan 220V-225V, dan *over voltage* tegangan diatas 225V. Alat ini menggunakan transformator variabel tegangan kapasitas 1500 VA dihubungkan dengan *carbon brush* yang digerakkan fleksibel menggunakan motor DC. Untuk membaca nilai tegangan dan arus dari keluaran transformator variabel tegangan menggunakan sensor PZEM-004T, lalu sensor PZEM-004T mengirim data ke Arduino. Setelah diolah data tegangan dan arus langsung di transfer ke LCD TFT secara *real time*. Pada percobaan pertama terjadi kondisi *over voltage* sebesar 231 V dan memiliki arus sebesar 0,14 A lalu di stabilkan oleh driver motor menjadi 222 V dan memiliki arus sebesar 0,14 A. Pada percobaan kedua terjadi kondisi *normal voltage* sebesar 221 V dan memiliki arus sebesar 1,35 A lalu pada kondisi ini driver motor tidak bergerak karena sudah pada kondisi stabil yaitu diantara 220 V – 225 V. Pada percobaan terakhir terjadi kondisi *under voltage* sebesar 215 V dan memiliki arus sebesar 1,39 A lalu di stabilkan oleh driver motor menjadi 222 V dan memiliki arus sebesar 1,37 A.

Kata Kunci: Over-Under Voltage, Arduino Uno, PZEM-004T, Driver Motor, Relay

Abstract

Electrical disturbances in consumers result in inhibition of everyday human activities. The disturbances that occur are from various aspects. One of the disturbances that occurs is the *Over-Under Voltage* interference in the power installation. This interference if connected to electrical equipment will affect the performance of the electrical equipment and reduce the life of the equipment. For this reason, a Voltage Stabilizer Prototype was made to Overcome *Over-Under Voltage* Interference in Power Installation Using Arduino. This tool shows 3 conditions, namely *under voltage* where the voltage is below 220V, *normal voltage* is 220V-225V, and *over voltage* is above 225V. This tool uses a variable voltage transformer with a capacity of 1500 VA connected to a carbon brush that is driven flexibly using a DC motor. To read the voltage and current values from the output of a variable voltage transformer using the PZEM-004T sensor, then the PZEM-004T sensor sends data to Arduino. After processing, the voltage and current data are transferred directly to the TFT LCD in real time. In the first experiment, there was an *over voltage* condition of 231 V and a current of 0.14 A then stabilized by the motor driver to 222 V and a current of 0.14 A. In the second experiment, the *normal voltage* was 221 V and had a current of 1.35 A then in this condition the motor driver does not move because it is already in a stable condition, which is between 220 V - 225 V. In the last experiment there was an *under voltage* condition of 215 V and a current of 1.39 A then stabilized by the motor driver to become 222 V and has a current of 1.37 A.

Keywords: Over-Under Voltage, Arduino Uno, PZEM-004T, Driver Motor, Relay

PENDAHULUAN

Semua aktivitas kebanyakan manusia tidak pernah terlepas dari penggunaan energi listrik. Kebutuhan energi listrik yang diperlukan oleh masyarakat di seluruh Indonesia akan terus meningkat, sehingga diperlukan suplai tenaga

listrik yang tercukupi. Tercatat, penjualan listrik PLN tahun 2018 menghabiskan sekitar 173 TWH (*Terra Watt Hour*) dibandingkan dengan tahun 2017 yang hanya menghabiskan 165,1 TWH atau terjadi kenaikan sekitar 4,87%. Selain dari pasokan listrik yang tercukupi, masyarakat juga membutuhkan kestabilan listrik dan

keamanan dari sebuah listrik itu sendiri. Sehingga mengurangi kemungkinan terjadinya sebuah gangguan. Maka dari itu PT. PLN sebagai penyuplai kelistrikan yang ada di Indonesia dituntut untuk memberikan kelistrikan yang stabil dan juga handal.

Adanya gangguan dalam suplai listrik dapat mempengaruhi bahkan merusak suatu sistem tenaga listrik. Gangguan yang terjadi dapat berbagai macam antara lain adalah lonjakan atau penurunan tegangan listrik (*over/under voltage*) (Arsyad, 2019). Gangguan tegangan lebih (*over voltage*) adalah gangguan sistem yang sering terjadi pada sistem jaringan terendah (Firmansyah, 2016). Gangguan tegangan kurang (*under voltage*) adalah gangguan sistem yang sering terjadi pada sistem jaringan menengah. Kedua gangguan ini jika terhubung pada peralatan listrik akan mempengaruhi kinerja peralatan listrik tersebut dan mengurangi umur peralatan tersebut. Untuk mengatasi gangguan *over voltage* dan *under voltage* dibuatlah alat *Prototype* Penstabil Tegangan Untuk Mengatasi Gangguan *Over-Under Voltage* Menggunakan Arduino. Alat ini menggunakan transformator variabel tegangan kapasitas 1500 VA dihubungkan dengan *carbon brush* yang digerakkan fleksibel menggunakan motor DC. Ketika ada gangguan *under voltage* (Tegangan kurang dari nominal), sensor tegangan PZEM-004T mendeteksi gangguan tersebut dan arduino sebagai pengolah data untuk mengirim perintah ke driver motor DC untuk mengatur putaran motor DC tersebut berputar ke arah kanan untuk menaikkan tegangan melalui transformator variabel tersebut sampai ke tegangan nominal. Jika ada gangguan *over voltage* (Tegangan lebih dari nominal), motor DC tersebut berputar ke arah kiri untuk menurunkan tegangan melalui transformator variabel tersebut sampai ke tegangan nominal.

Prototype ini dilengkapi dengan sensor PZEM-004T, relay untuk mengatur indikator lampu LED, LCD TFT ukuran 16 x 2 sebagai penampilan data tegangan dan arus pada output transformator variable tegangan dan juga lampu LED sebagai indikator keadaan tegangannya.

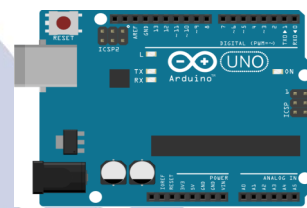
KAJIAN TEORI

Arduino Uno

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328 (Widarsono, 2019). Arduino dirancang untuk memudahkan penggunaan alat elektronik dalam berbagai bidang. *Hardware* Arduino memiliki prosesor Atmel AVR dan *software*nya yang dirancang untuk memiliki Bahasa peromagaman sendiri. Papan mikrokontroler Arduino Uno itu sendiri adalah IC (integrated circuit) yang bisa dengan mudah diprogram menggunakan komputer menggunakan bahasa pemrograman yang memiliki kemiripan seperti bahasa pemrograman C. Pemrograman Arduino Uno dapat di

akses menggunakan Arduino IDE. Tujuan dibekali program arduino ini adalah memudahkan rangkaian elektronik dapat membaca input, memproses input arduino tersebut dan akan menghasilkan keluaran sesuai yang diinginkan. Mikrokontroler Arduino bertugas sebagai ‘otak’ yang mengendalikan masukan (*input*), proses, dan keluaran (*output*) pada rangkaian elektronik. Pada Gambar 1.

Berikut spesifikasi Arduino Uno yaitu : Tegangan *input* 7-12V, 14 digital pin *input* / output (dimana 6 pin dapat digunakan sebagai output PWM), *Flash Memory* 32 KB (ATmega328), 6 *input* analog / ADC (Analog to Digital Converter), kristal 16 MHz, Dilengkapi dengan koneksi USB tipe B, *header* ICSP dan tombol reset.



Gambar 1. Arduino

(Sumber : <https://www.nyebarilmu.com/mengenal-apa-itu-arduino-uno/>)

Sensor PZEM-004T

PZEM-004T adalah sebuah modul sensor multifungsi yang berfungsi untuk mengukur tegangan dan juga arus pada sebuah saluran listrik. Modul ini sudah dilengkapi dengan sensor tegangan yang terintegrasi. PZEM-004T juga dilengkapi dengan *current transformer* model *split core* digunakan untuk mengukur arus. Penggunaan *split core* memiliki keunggulan dan kemudahan penggunaannya karena bisa langsung dipasang pada kabel jaringan listrik yang sudah terpasang tanpa harus memutus kabel listrik tersebut. Dalam penggunaannya alat ini khusus digunakan untuk *in door* dan beban yang terpasang tidak diperbolehkan melebihi daya yang sudah ditetapkan. Pada Gambar 2.

Berikut spesifikasi PZEM-004T : Suhu operasi : -40 C + 70 C, memiliki toleransi error 1.5%, Tegangan kerja pada arduino 5V, Perbandingan rasio keluaran sesuai tegangan sumber.



Gambar 2. Sensor PZEM-004T

(Sumber : <https://mikroavr.com/sensor-pzem-004t-arduino/>)

Relay

Relay adalah komponen elektronika yang berupa saklar. Relay terdiri dari 5 komponen penting yaitu elektromagnet yang berupa koil, *armature*, *spring*, *switch*, dan kontak saklar yang dapat dioperasikan semi otomatis serta dapat dihubungkan dengan arduino. Komponen elektronika ini menggunakan prinsip elektromagnetik untuk menggerakkan saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik yang memiliki tegangan lebih tinggi. Relay memiliki 2 opsi, yaitu *nomally open* yang berarti ketika relai mendapatkan suplai energi listrik akan ON dan *normally close* yang berarti ketika relay mendapat suplai energi listrik akan OFF. Diantara aplikasi relay yang dapat ditemui diantaranya adalah relay sebagai kontrol ON/OFF beban dengan sumber tegang berbeda (Turang, 2015). Pada Gambar 3

Berikut spesifikasi relay tersebut : tegangan *input* 5V, memiliki dimensi 48mm x 18mm x 20mm, kontak NO, NC, dan com, indikator tegangan led hijau, indikator relay led merah.



Gambar 3. Relay

(sumber : <https://sunupradana.info/tnr/2017/07/15/modul-dua-relay/>)

Driver Motor IC L298

Driver Motor L298N adalah sebuah modul yang digunakan untuk mengendalikan arah putaran motor DC. Pada driver ini terdapat komponen IC L298. IC driver L298 memiliki kemampuan menggerakkan motor DC sampai arus 2A dan tegangan maksimum 40 volt DC untuk satu kanalnya. (Adriansyah dan Hidyatama, 2013). Pada IC L298 terdapat transistor logic dengan gerbang nand yang berfungsi sebagai penentu arah putaran beban motor DC ataupun Stepper. Modul ini bisa digunakan untuk motor DC yang tegangan *input*-nya lebih dari 12 volt dan juga kuat untuk beban-beban yang cukup besar tegangan *input*-nya. Kelebihan dari driver motor L298N ini adalah cukup presisi dalam mengontrol motor (Widiarto, 2018). Pada Gambar 4

Berikut spesifikasi Driver motor ic l298 : Tegangan input minimal antara 5V-35V, input arus antara 0-36mA, Output arus pada A atau B yaitu 2A, memiliki dimensi 43 x 43 x 26mm, Menggunakan IC L298N (Double H bridge Drive Chip).



Gambar 4. Driver Motor IC L298

(Sumber : <https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/>)

LCD 16 x 2

LCD (*Liquid Crystal Display*) adalah suatu jenis media display yang menggunakan kristal cair untuk menghasilkan sebuah tampilan. LCD pada dasarnya terdiri dari 2 bagian utama yaitu bagian *backlight* dan bagian kristal cair. LCD tidak memancarkan cahaya sama sekali, LCD hanya merefleksikan dan menstranmisikan cahaya yang melewatinya. Oleh karena itu LCD membutuhkan cahaya *backlight* yang pada umunya berwarna putih sebagai sumber cahayanya. Kristal cair (*Liquid Crystal*) sendiri adalah suatu cairan organik yang terletak diantara dua lembar kaca dengan elektroda transparan indium oksida dalam bentuk tampilan seven-segment yang permukaannya transparan dan konduktif elektroda pada kaca belakang. Prinsip kerja LCD ialah *backlight* yang berwarna putih akan memberikan cahaya pada kristal cair, lalu kristal cair tersebut akan menyaring *backlight* yang diterimanya dan merefleksikan sesuai dengan sudut yang diinginkan sehingga menghasilkan warna yang dibutuhkan. Pada Gambar 5

Berikut Spesifikasi LCD 16 x 2 : Bisa dihubungkan Arduino maupun mikrokontroler lainnya, tegangan kerja 5V , memiliki sistem I2C sehingga mempersingkat pemrograman, dilengkapi dengan pengatur lampu dan kontras kecerahan, dimensi ukuran: 41.5x19x15.3mm.



Gambar 5. LCD 16 x 2

(sumber : <https://www.nyebarilmu.com/cara-mengakses-modul-display-lcd-16x2/>)

Arduino IDE

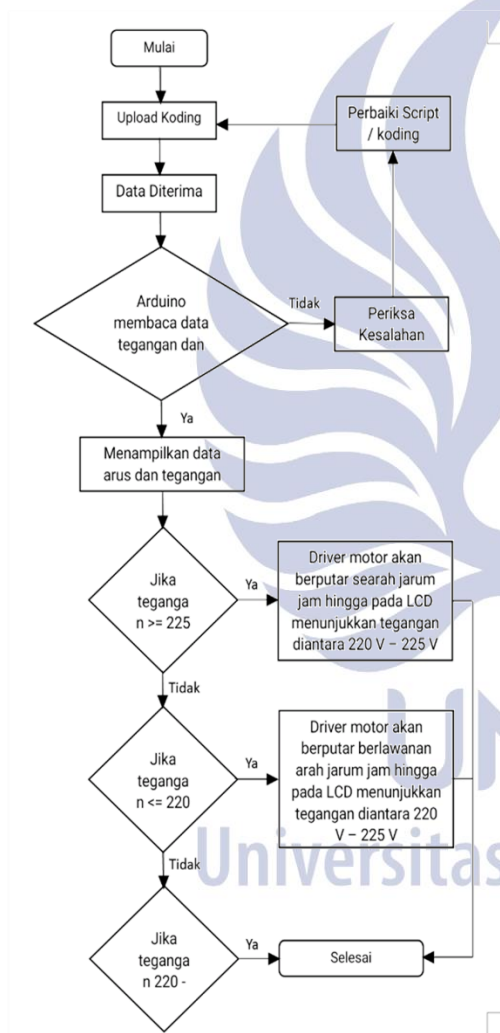
IDE secara bahasa adalah lingkungan terintegrasi yang dilakukan untuk melakukan pengembangan. Arduino IDE dibuat dari Bahasa pemrograman java. Arduino IDE juga dilengkapi Bahasa C/C++ untuk memprogram Arduino agar lebih mudah. Program yang ditulis dengan menggunakan Arduino IDE disebut juga dengan *sketch*.

Metode Penelitian

Metode yang digunakan adalah metode percobaan atau eksperimen. Metode ini adalah sesuatu yang dibutuhkan dalam bidang keilmuan untuk melakukan sebuah perkembangan teknologi maupun lainnya yang bisa dimanfaatkan dan dipelajari oleh masyarakat secara aman, nyaman, dan terpercaya. Dalam pembelajarannya melibatkan siswa dan juga mahasiswa yang ingin mempelajari bidang tersebut untuk membuktikan hasil percobaannya. Pada metode penelitian ini meliputi berbagai rancangan *software*.

Perancangan Pembuatan Software

Diagram line pada pembuatan *software* ditunjukkan seperti gambar 7 berikut:

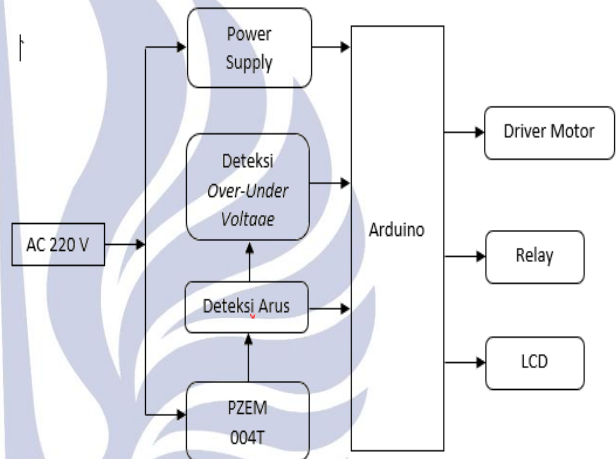


Gambar 7. Flowchart Software

Pembuatan perangkat lunak (*software*) berfungsi sebagai *support* dalam perangkat keras karena semua sistem yang mengatur *hardware* adalah *software*. *Software* utama yang dipakai dalam *prototype* ini adalah arduino, sensor arus, sensor tegangan, relai, dan driver motor. Langkah kerja yang dilakukan pada arduino uno yaitu,

Langkah Pertama dilakukan dengan pembuatan program untuk pembacaan sensor arus, sensor tegangan, relai, LCD 16 x 2, dan driver motor. Langkah kedua menampilkan hasil data sensor tegangan dan sensor arus pada LCD. Langkah ketiga driver motor bergerak untuk menstabilkan dengan data yang sudah terdeteksi pada sensor tegangan, jika tegangan *under* maka driver motor akan berputar berlawanan arah jarum jam hingga mencapai tegangan diantara 220 V – 225 V, jika tegangan *over* maka driver motor akan berputar searah jarum jam hingga mencapai tegangan diantara 220 V – 225 V dan, jika diantara 220 V – 225 V maka driver motor tidak akan berputar

Diagram blok pada pembuatan *software* ditunjukkan seperti gambar 8 berikut:



Gambar 8. Diagram Blok Software

Secara garis besar prinsip kerja sistem ini berdasarkan blok diagram dapat dijelaskan yaitu, tegangan input masuk ke sensor PZEM-004T dan *power supply* untuk menghidupkan Arduino, lalu sensor PZEM-004T mendeteksi arus dan juga tegangan yang akan dikirimkan ke Arduino. Dengan adanya data yang sudah terkirim maka, Arduino mengolah data dari sensor untuk dikirim ke relay, LCD, dan juga driver motor. Relay berfungsi untuk menghidupkan lampu dengan warna hijau kondisi *normal voltage*, warna merah kondisi *under voltage*, dan warna kuning kondisi *over voltage*. LCD berfungsi untuk menampilkan hasil arus dan tegangan yang sudah di rekam oleh sensor secara *real time*. Driver motor berfungsi untuk menstabilkan tegangan hingga tegangan berada di antara 220 V – 225 V.

HASIL DAN PEMBAHASAN

Definisi Operasional Alat

Perangkat *prototype* penstabil tegangan merupakan sebuah *prototype* yang dapat menstabilkan tegangan dikala tegangan terjadi *under* maupun *over*. *Prototype* ini

menggunakan transformator berkapasitas 1500 VA yang digabung dengan *carbon brush* beserta driver motor sebagai fungsi utamanya yaitu menstabilkan tegangan tidak lupa juga dengan adanya mikrokontroler arduino uno sebagai otak dari *prototype* ini dan juga beserta beberapa sensor sebagai peran pembantu agar *prototype* ini dapat bekerja terhadap fungsi utamanya dengan maksimal. *Prototype* ini memiliki fungsi yang sangat berguna untuk menghindari terjadinya *under* dan *over voltage* pada jaringan sistem tegangan rendah dengan adanya hal ini maka akan terjadi pemanisiran kerusakan alat.

Cara kerja *prototype* ini diawali dengan pembacaan sensor tegangan dan arus dengan dibantu arduino lalu arduino mengirim hasil data tersebut ke LCD 16 x 2 yang akan di tampilkan secara *real time*. Setelah hasil tegangan dan arus ditampilkan maka driver motor akan menstabilkan tegangan. Jika terdeteksi *under voltage* driver motor akan berputar berlawanan dengan arah jarum jam, lalu jika terdeteksi *over voltage* driver motor akan berputar searah jarum jam dan jika terdeteksi diantara 220 V – 225 V maka motor tidak akan berputar karena tegangan sudah dianggap stabil.

Hasil

Hasil dari tugas akhir ini adalah perancangan program Arduino yang akan dimasukkan ke dalam *prototype* penyetabil tegangan untuk mengatasi gangguan *over-under voltage* pada instalasi daya menggunakan Arduino. Pada pembuatan tugas akhir berjudul perancangan program *prototype* penyetabil tegangan untuk mengatasi gangguan *over-under voltage* pada instalasi daya menggunakan Arduino terdiri dari beberapa komponen, yaitu: Arduino, sensor PZEM-004T, LCD, modul IC L298N, dan relay.

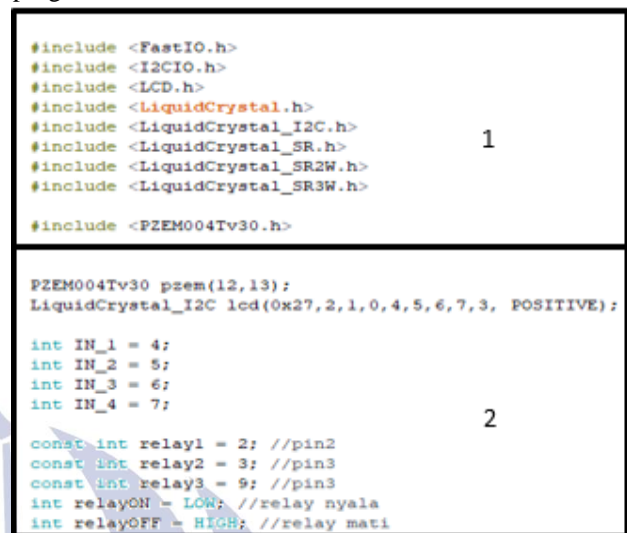
Secara garis besar sistem kerjanya adalah jika, sensor PZEM-004T mendeteksi tegangan <220 atau dikatakan *under voltage*, maka driver motor akan memerintahkan motor DC berputar berlawanan jarum jam untuk menaikkan tegangan hingga kembali ke tegangan nominal dan jika, sensor PZEM-004T mendeteksi tegangan >220 atau dikatakan *over voltage*, maka driver akan memerintahkan motor DC untuk berputar searah jarum jam untuk menurunkan tegangan hingga kembali ke tegangan nominal.

Berikut adalah perancangan program Arduino untuk sistem pengendalian *prototype* penyetabil tegangan pada instalasi.

A) Perancangan program deklarasi variabel

Bagian deklarasi awal digunakan untuk mendeskripsikan variabel-variabel yang akan digunakan dalam program utama dan juga untuk menambahkan file-

file pada program yang dibutuhkan untuk menjalankan program utama. Pada Gambar 9.



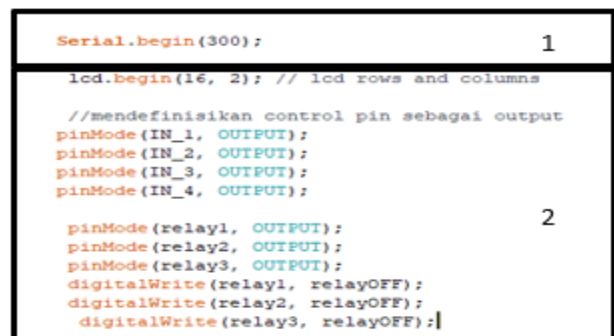
Gambar 9. Screenshot Command Prompt Arduino IDE Perancangan Program Deklarasi Variabel

Penjelasan gambar 9:

1. Deklarasi fungsi *library* sensor PZEM-004T, dan LCD yang dimana di dalam *library* tersebut berisi tentang kumpulan kode yang dapat mengeksekusi perintah-perintah tertentu berdasarkan fungsi dan prosedur yang terbangun oleh file *header*-nya.
2. Deklarasi yang berfungsi sebagai penetapan pin pada penggunaan sensor PZEM-004T yang di tetapkan pada pin 12, dan 13 pada Arduino ,lalu driver motor yang di tetapkan pada pin 4, 5, 6, dan 7 pada Arduino, lalu relay yang di tetapkan pada pin 2, 3, dan 9 pada Arduino.

B) Perancangan program fungsi void setup

Bagian setup digunakan untuk menginisialisasi variabel, mengatur mode pin pada *board*, mengatur *timer*, mengatur *baudrate serial port* dan sebagainya. Bagian setup hanya dijalankan sekali saja yaitu pada saat awal program dijalankan atau ketika program di reset. Pada Gambar 10.



Gambar 10. Screenshot Command Prompt Arduino IDE Perancangan Program Void Setup

Penjelasan gambar 10 :

1. Inisialisasi *baudrate serial port* untuk LCD
2. Inisialisasi pin relay sebagai output yang dihubungkan pada pin digital Arduino dengan kondisi relay NO

C) Perancangan program void loop

Bagian *loop* adalah fungsi utama program yang akan dijalankan berulang-ulang. Pada fungsi *loop* dibawah terdapat fungsi untuk membaca sensor PZEM-004T lalu ditampilkan pada LCD, fungsi relay, dan fungsi untuk menjalankan motor DC yang akan dijalankan berulang-ulang untuk menstabilkan tegangan jika terjadi *over voltage* atau *under voltage*. Pada Gambar 11.

```
void loop() {
  float voltage = pzem.voltage();
  if(voltage != 0){
    Serial.print("Voltage: "); Serial.print(voltage); Serial.println("");
    lcd.setCursor(0,0);
    lcd.print("V= ");
    lcd.setCursor(2,0);
    lcd.print(voltage);
  } else {
    Serial.println("Error reading voltage");
  }
  float current = pzem.current();
  if(current != 0){
    Serial.print("Current: "); Serial.print(current); Serial.println("");
    lcd.setCursor(9,0);
    lcd.print("I= ");
    lcd.setCursor(11,0);
    lcd.print(current);
  } else {
    Serial.println("Error reading current");
  }
  Serial.println();
  delay(300);

  if (voltage >225){
    digitalWrite(relay1, relayON);
  }
  if (voltage <225){
    digitalWrite(relay1, relayOFF);
  }

  if (voltage <220){
    digitalWrite(relay2, relayON);
  }
  if (voltage >220){
    digitalWrite(relay2, relayOFF);
  }
  if (voltage >220){
    digitalWrite(relay3, relayON);
  }
  if (voltage <225){
    digitalWrite(relay3, relayON);
  }
  if (voltage <220){
    digitalWrite(relay3, relayOFF);
  }
  if (voltage >225){
    digitalWrite(relay3, relayOFF);
  }
}
```

```
//Putar Mesin searah jarum jam
if (voltage >225){
  digitalWrite(IN_1, HIGH);
  digitalWrite(IN_2, LOW);

  delay(1000);
  //Untuk mesin A
  digitalWrite(IN_1, HIGH);
  digitalWrite(IN_2, HIGH);
  delay(1500);
}

else if (voltage <225){
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, LOW);

  delay(1500);
  //Untuk mesin A
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, LOW);
  delay(1500);
}
else if (voltage >220) (voltage =225);{
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, LOW);
  delay(1500);
  //Untuk mesin A
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, LOW);
  delay(1500);
}

//Putar Mesin berlawanan arah jarum jam
if (voltage <220){
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, HIGH);

  delay(1000);
  //Untuk mesin A
  digitalWrite(IN_3, HIGH);
  digitalWrite(IN_4, HIGH);
  delay(1500);
}

else if (voltage >220){
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, LOW);

  delay(1500);
  //Untuk mesin A
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, LOW);
  delay(1000);
}
else if (voltage >220) (voltage <225);{
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, LOW);

  delay(1500);
  //Untuk mesin A
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, LOW);
  delay(1000);
}

//PETERANGAN LCD
if(voltage >225){
  Serial.print("OVERVOLTAGE: "); Serial.print(voltage); Serial.println("");
  lcd.setCursor(2,1);
  lcd.print("OVERVOLTAGE");
  delay(300);
}

if(voltage <225){
  Serial.print("UNDERVOLTAGE: "); Serial.print(voltage); Serial.println("");
  lcd.setCursor(2,1);
  lcd.print("UNDERVOLTAGE");
  delay(300);
}

if(voltage >220)
if(voltage <225){
  Serial.print("NORMAL VOLTAGE: "); Serial.print(voltage); Serial.println("");
  lcd.setCursor(1,1);
  lcd.print("NORMAL VOLTAGE");
  delay(300);
}
}
```

Gambar 11. Screenshot Command Prompt Arduino IDE Perancangan Program Void Loop

Penjelasan dari gambar 11:

1. Pada koding ini berfungsi sebagai penempatan hasil sensor PZEM-004T yang berupa arus dan juga tegangan pada LCD 16 x 2.
2. Pada koding ini berfungsi sebagai mengatur *relay* pada saat kondisi *over voltage* maka, *relay* 1 akan menyala dan *relay* 2 dan *relay* 3 mati, lalu pada saat kondisi *under voltage* maka, *relay* 2 akan menyala dan *relay* 1 dan *relay* 3 mati, yang terakhir pada kondisi normal *voltage* maka, *relay* 3 akan menyala dan *relay* 1 dan *relay* 2 mati.
3. Pada koding ini berfungsi sebagai mengatur driver motor pada saat kondisi *over voltage* maka, driver motor akan berputar searah jarum jam untuk menstabilkan tegangan hingga mencapai tegangan diantara 220 V – 225 V. Pada saat kondisi *under voltage* maka, driver motor akan berputar berlawanan jarum jam untuk menstabilkan tegangan hingga mencapai tegangan diantara 220 V – 225 V dan pada saat kondisi normal *voltage* maka, driver motor tidak akan berputar.
4. Pada koding ini berfungsi sebagai penempatan keterangan bahwa tegangan mengalami normal *voltage*, *under voltage*, dan *over voltage* yang akan tampil di LCD bagian bawah.

Hasil Pengujian Software

```

10:51:35.696 -> Voltage: 231.10
10:51:36.489 -> Current: 0.14
10:51:36.489 -> OVERVOLTAGE: 231.40
10:51:46.873 -> Voltage: 226.50
10:51:47.703 -> Current: 0.14
10:51:47.703 -> OVERVOLTAGE: 226.50
10:51:58.572 -> Voltage: 222.00
10:51:59.637 -> Current: 0.14
10:51:59.637 -> NORMAL VOLTAGE: 222.00
    
```

Gambar 12. Hasil *Real Time* Koding Pada Kondisi *Over Voltage* Hingga Menjadi Normal *Voltage*

Pada gambar 12 menyatakan hasil tegangan dan arus yang sudah terdeteksi oleh sensor PZEM-004T dengan tegangan sebesar 231.10 V dan arus sebesar 0.14 A sehingga tegangan terdeteksi sebagai *over voltage*, lalu Arduino sebagai pengolah data mengirimkan kepada driver motor agar driver motor menstabilkan tegangannya secara *real time* hingga menjadi 222 V dan arus sebesar 0.14 A seperti gambar diatas.

```

11:23:01.549 -> Voltage: 221.80
11:23:01.549 -> Current: 1.35
11:23:02.097 -> NORMAL VOLTAGE: 221.80
11:23:13.541 -> Voltage: 221.80
11:23:13.541 -> Current: 1.35
11:23:14.091 -> NORMAL VOLTAGE: 221.80
11:23:25.551 -> Voltage: 222.00
11:23:25.551 -> Current: 1.34
11:23:26.102 -> NORMAL VOLTAGE: 222.00
    
```

Gambar 13. Hasil *Real Time* Koding Pada Kondisi Normal *Voltage*

Pada gambar 13 menyatakan hasil tegangan dan arus yang sudah terdeteksi oleh sensor PZEM-004T dengan tegangan sebesar 221.80 V dan arus sebesar 1.35 A sehingga tegangan terdeteksi sebagai normal *voltage* dan pada gambar diatas tidak terjadi perubahan tegangan yang terlalu signifikan dikarenakan hasilnya sudah menandakan tegangan yang stabil yaitu diantara 220 V – 225 V.

```

11:22:13.359 -> Voltage: 215.60
11:22:14.202 -> Current: 1.39
11:22:14.202 -> UNDERVOLTAGE: 215.60
11:22:25.087 -> Voltage: 218.70
11:22:25.916 -> Current: 1.39
11:22:25.916 -> UNDERVOLTAGE: 218.70
11:22:36.767 -> Voltage: 221.80
11:22:37.611 -> Current: 1.39
11:22:37.611 -> NORMAL VOLTAGE: 221.80
    
```

Gambar 14. Hasil *Real Time* Koding Pada Kondisi *Under Voltage* Hingga Menjadi Normal *Voltage*

Pada gambar 14 menyatakan hasil tegangan dan arus yang sudah terdeteksi oleh sensor PZEM-004T dengan tegangan sebesar 215.60 V dan arus sebesar 1.39 A sehingga tegangan terdeteksi sebagai *under voltage*, lalu Arduino sebagai pengolah data mengirimkan kepada driver motor agar driver motor menstabilkan tegangannya secara *real time* hingga menjadi 221.80 V dan arus sebesar 1.37 A seperti gambar diatas.

Hasil Error Tegangan dan Arus Kondisi *Over Voltage*

No	Tegangan Multimeter	Tegangan PZEM-004T	Error (V)	Error (%)
1	232	231	1	0.43
2	227	226	1	0.44
3	223	222	1	0.45

Tabel 1. Perbandingan Tegangan Antara Multimeter Dengan *Software* Kondisi *Over Voltage*

Hasil perbandingan antara 3 kali pengukuran tegangan dengan multimeter dan juga *software*

menghasilkan perbandingan 1 V dan *prosentase* yang berbeda yaitu 0.43%, 0.44%, dan 0.45% pada Tabel 1.

No	Arus Multimeter	Tegangan PZEM-004T	Error (A)	Error (%)
1	0.15	0.14	0.01	6.67
2	0.15	0.14	0.01	6.67
3	0.15	0.14	0.01	6.67

Tabel 2. Perbandingan Arus Antara Multimeter Dengan *Software* Kondisi *Over Voltage*

Hasil perbandingan antara 3 kali pengukuran arus dengan multimeter dan juga *software* menghasilkan perbandingan 0.1 A dan *prosentase* yang sama yaitu 6.67% seperti pada Tabel 2.

Hasil Error Tegangan dan Arus Kondisi Normal Voltage

No	Tegangan Multimeter	Tegangan PZEM-004T	Error (V)	Error (%)
1	222	221	1	0.45
2	222	221	1	0.45
3	223	222	1	0.45

Tabel 3. Perbandingan Tegangan Antara Multimeter Dengan *Software* Kondisi Normal Voltage

Hasil perbandingan antara 3 kali pengukuran tegangan dengan multimeter dan juga *software* menghasilkan perbandingan 1 V dan *prosentase* yang sama yaitu, 0.45% seperti pada Tabel 3.

No	Arus Multimeter	Tegangan PZEM-004T	Error (A)	Error (%)
1	1.44	1.35	0.1	6.94
2	1.44	1.35	0.1	6.94
3	1.43	1.34	0.1	6.99

Tabel 4. Perbandingan Arus Antara Multimeter Dengan *Software* Kondisi Normal Voltage

Hasil perbandingan antara 3 kali pengukuran arus dengan multimeter dan juga *software* menghasilkan perbandingan 0.1 A dan *prosentase* yang sama berbeda yaitu 6.94% dan 6.99% seperti pada Tabel 4.

Hasil Error Tegangan dan Arus Kondisi Under Voltage

No	Tegangan Multimeter	Tegangan PZEM-004T	Error (V)	Error (%)
1	216	215	1	0.46
2	219	218	1	0.45
3	223	222	1	0.45

Tabel 5. Perbandingan Tegangan Antara Multimeter Dengan *Software* Kondisi *Over Voltage*

Hasil perbandingan antara 3 kali pengukuran tegangan dengan multimeter dan juga *software* menghasilkan perbandingan 1 V dan *prosentase* yang berbeda yaitu, 0.46% dan 0.45% seperti pada Tabel 5.

No	Arus Multimeter	Tegangan PZEM-004T	Error (A)	Error (%)
1	1.49	1.39	0.1	6.71
2	1.49	1.39	0.1	6.71
3	1.47	1.37	0.1	6.80

Tabel 6. Perbandingan Arus Antara Multimeter Dengan *Software* Kondisi *Over Voltage*

Hasil perbandingan antara 3 kali pengukuran arus dengan multimeter dan juga *software* menghasilkan perbandingan 0.1 A dan *prosentase* yang sama berbeda yaitu 6.71% dan 6.80% seperti pada Tabel 4.

PENUTUP

Simpulan

Berdasarkan pelaksanaan dari hasil kegiatan pemrograman *prototype* penstabil tegangan untuk mengatasi gangguan *over – under voltage* pada instalasi daya berbasis Arduino yang telah dikerjakan untuk memenuhi mata kuliah tugas akhir, dapat ditarik beberapa kesimpulan bahwa pemrograman untuk *prototype* penstabil tegangan untuk mengatasi gangguan *over-under voltage* pada instalasi daya berbasis Arduino menggunakan Bahasa C pada *software* Arduino dan unjuk kerja dari pemrograman untuk *prototype* penstabil tegangan untuk mengatasi gangguan *over-under voltage* berbasis Arduino yaitu, setelah program di *upload* ke Arduino. Program akan berjalan sesuai dengan rencana atau bisa dikatakan berhasil dan menghasilkan tegangan pada kondisi *under voltage* 215.80 V lalu di stabilkan menjadi 221.80 V dengan arus awal sebesar 1.39 A lalu ketika stabil berubah menjadi 1.37 A. Pada kondisi normal voltage tegangan tidak terjadi perubahan yang signifikan yaitu 221.80 V menjadi 222.00 karena tegangan sudah stabil dan memiliki arus sebesar 1.35 A lalu ketika tegangan berubah menjadi 1.34 A. Percobaan terakhir pada kondisi *over voltage* memiliki tegangan awal 231.10 V lalu di stabilkan menjadi 222.00 V dan memiliki arus awal sebesar 0.14 dan ketika tegangan stabil arus tetap sama.

Saran

Prototype ini masih bisa lebih dikembangkan lagi dengan cara ditambahkan lebih banyak sensor lagi ataupun *prototype* ini bisa juga digabungkan dengan *prototype* yang lainnya sehingga memiliki fungsi yang lebih maksimal dan bisa diperbanyak atau diperjual belikan pada masyarakat setempat yang membutuhkan. Fungsi dari *prototype* ini pasti sangat dibutuhkan karena dapat membuat alat

kelistrikan rumah tangga menjadi sedikit lebih awet dengan adanya sistem penstabil tegangan pada *prototype* ini.

DAFTAR PUSTAKA

- Andriansyah, Andi dan Oka Hidyatama. 2013. *Rancang Bangun Prototype Elevator Menggunakan Mikrokontroller Arduino Atmega 328p*. Manado: Universitas Sam Ratulangi.
- Arsyad, Billy Mahdianto. 2019. *Perancangan Sistem Kontrol Over/Under Voltage Relay Berbasis Mikrokontroler Pada Saluran Tegangan 220VAC*. Semarang: Universitas Diponegoro.
- Firmansyah, Riza Agung. 2016. *Perancangan dan Pembuatan Alat Proteksi Gangguan Tegangan Lebih Berbasis Mikrokontroler*. Surabaya: Institut Teknologi Adhi Tama Surabaya.
- Muslim, Supari. 2009. *Teknik Perencanaan dan Pemasangan Instalasi Listrik*. Surabaya: Direktorat Pembinaan Sekolah Menengah Kejuruan.
- Syahwil, Muhammad. 2013. *"Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino"*. Yogyakarta : Edisi-1.
- Turang, Daniel Alexander Octavianus. 2015. *Pengembangan Sistem Relay Pengendalian dan Penghematan Pemakaian Lampu Berbasis Mobile*. Bontang : Sekolah Tinggi Teknologi Bontang
- Widarsono, Kuku. 2019. *Perlindungan Relay Over/Under Voltage dan Ketidakseimbangan Tegangan Berdasarkan Visual Basic Menggunakan Arduino*. Sampang: Politeknik Negeri Madura.
- Widiarto, Yosua Dwi. 2018. *Sistem Penggerak Robot Beroda Vacuum Cleaner Berbasis Mini Computer Raspberry pi*. Manado: Universitas Sam Ratulangi.

